



A12.4 The TCP/IP stack and protocols

Networks and the web · A level · OCR H446 1.3.3, AQA 7517 4.9.4.1,
Eduqas A500QS 2.2 · about 30 min

Name _____

Class _____

Date _____

What this lesson is about

The four layers, encapsulation, ports and sockets, and HTTP, HTTPS, FTP, SMTP, POP3 and SSH.

- Four layers
- Ports and sockets
- The application protocols
- A packet of your own

Questions

6 marks in all

1. Put the layers of the TCP/IP stack in order, from the top (nearest the application) to the bottom. [1 mark]

Number the lines 1 to 4 to put them in the right order.

- ___ Link
- ___ Network
- ___ Transport
- ___ Application

2. Which layer adds the source and destination port numbers? [1 mark]

- ☐ A Transport
- ☐ B Application
- ☐ C Network
- ☐ D Link

3. What is a socket? [1 mark]

- ☐ A The combination of an IP address and a port number, one endpoint of a connection
- ☐ B A physical connector on a network card
- ☐ C A 48-bit hardware address
- ☐ D The list of rules in a firewall

4. On which port does a web server listen for HTTPS by default? [1 mark]

-
5. Which protocol does an email client use to download messages from its mail server, usually deleting them from the server? [1 mark]

- ☐ A POP3
- ☐ B SMTP
- ☐ C FTP
- ☐ D SSH

6. As a packet crosses several routers on the Internet, which addresses are replaced at every hop? [1 mark]
- ☐ A The MAC addresses in the link layer header
 - ☐ B The IP addresses in the network layer header
 - ☐ C The port numbers in the transport header
 - ☐ D None of them change

The task: wrap it and unwrap it

The robot at 192.168.4.23 asks the Server at 192.168.4.1 for its battery level. Write `checksum(text)`: its input is any string; it returns the sum of the character codes (`ord`) of every character in `text`, modulo 256, as two uppercase hexadecimal digits (`format(n, "02X")`). 1. Build and `send` one packet: the body `<source IP>,<destination IP>,<source port>,<destination port>,<data>` with source port 49152, destination port 80 and data `GET /battery`, then `*`, then the body's checksum. 2. The Server replies with a packet in the same format. Wait for it, checking `messages()` every 0.1 s. 3. Split off the checksum after the last `*`. If the body's checksum matches, print `checksum ok`, then `from socket <source IP>:<source port>`, then `data: <data>`. The data is everything after the fourth comma. If it does not match, print `checksum wrong`. The robot does not drive.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def checksum(text):
    return "00"

send("192.168.4.23,192.168.4.1,49152,80,GET /battery")
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/a12-4-tcp-ip-and-protocols/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Swap two characters in a body. Does the checksum notice? Why not, and what does that say about this checksum?
2. Add a TTL field to the packet and a router function that decrements it and drops the packet at 0.
3. Look up which port IMAP uses and why many people now use it instead of POP3.

